A SHADER BASED ADAPTATION

OF SELECTED SIXTEENTH CENTURY MAPS

A Thesis

by

SHAILA HAQUE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2008

Major Subject: Visualization Sciences

A SHADER BASED ADAPTATION

OF SELECTED SIXTEENTH CENTURY MAPS

A Thesis

by

SHAILA HAQUE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Carol LaFayette
Committee Members,	Richard Davison
	Katherine H. Weimer
Head of Department,	Tim McLaughlin

May 2008

Major Subject: Visualization Sciences

ABSTRACT

A Shader Based Adaptation of Selected Sixteenth Century Maps.

(May 2008)

Shaila Haque, B.S., Texas A&M University

Chair of Advisory Committee: Prof. Carol LaFayette

This research develops a technique focused on shading and texturing, with an emphasis on line work and color, to emulate the unique qualities of copperplate line-engraving from 16th century cartography. A visual analysis of selected maps determines the defining characteristics adapted for three-dimensional computer generated environments. The resulting work is presented in a short time-based animation.

for my family

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Prof. Carol LaFayette, and my committee members, Prof. Richard Davison and Ms. Katherine Weimer, for their guidance and support during the course of this research.

Also, I would like to thank the department faculty and staff for their help during my graduate studies. Thanks to my friends and fellow students for making my time in the Visualization Sciences program a memorable experience. I would like to thank Joseph Dellinger for his support, Mayank Singh and Eric Peden for their technical advice, and Elizabeth Nitsch for lending her artistic eye throughout this research.

Finally, I would like to thank my family for their unconditional love, encouragement, and support.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

“A map is a graphic document in which location, extent, and direction can be more precisely defined than by the written word; and its construction is a mathematical process strictly controlled by measurement and calculation. The completed map must nevertheless be drawn by the hand of the cartographer.” [1]

Early maps are adorned with elaborate, ornate details that are intertwined seamlessly with the necessary geographical information. Fantastical creatures and battle ships can be seen lurking in the swirling patterns of the water waves; houses sit atop hills and mountains dot the countryside. These maps have a distinctive visual style that can be considered works of art (see Figure 1).

Figure 1: Line-engraved map of *The New World* (1587) [1]

This thesis follows the style of *IEEE Transactions on Visualization and Computer Graphics*.

Artistic Intent

The goal of this thesis is to translate the style of two-dimensional (2D) maps to a three-dimensional (3D) environment, emphasizing the use of line, pattern, and color. In particular, this work will focus on decorated copper-engraved maps from the 16th century.

This research will produce an interpretation of copper plate line-engraving as a distinctive, non-photorealistic rendering technique. The final outcome will result in a short time-based work.

Objective

The objective of this thesis is to create a shader-based adaptation of 16th century decorated maps for a 3D environment. The process of intaglio as a printmaking technique was studied, with a focus on copper-engraving. The style of 16th century maps was explored, noting the use of line, pattern, texture, and color. This study concentrated specifically on the nautical sections of the map, and particular attention was given to the elements such as the ships, sea creatures, and water waves. Based on the visual analysis of the maps, a 3D modeled environment was shaded and lit, and additional details were incorporated with the use of texture mapping. A short time-based piece illustrates how the look of this distinctive 2D style translated to a 3D space.

CHAPTER II

BACKGROUND

Interpreting the look of copper plate line-engraving requires knowledge of printmaking techniques and understanding of the key characteristics of maps from this century.

Printmaking Techniques

Creating and transferring artistic designs onto paper or fabrics is a process known as printmaking. Printmaking techniques can be divided into three major categories: relief, intaglio, and surface. Map-making of the 16^{th} century makes use of the intaglio process, and more specifically, the line-engraving process used for copper plate printing.

"Intaglio" is from the Italian word *itagliare*, which translates as "cut into". A copper plate is incised by hand using a steel chisel-like tool called a burin. Engraving with a burin results in clean-edged, recessed lines in the copper plate, as illustrated in Figure 2. The number, placement, and direction of these lines will create the tones and textures of the print.

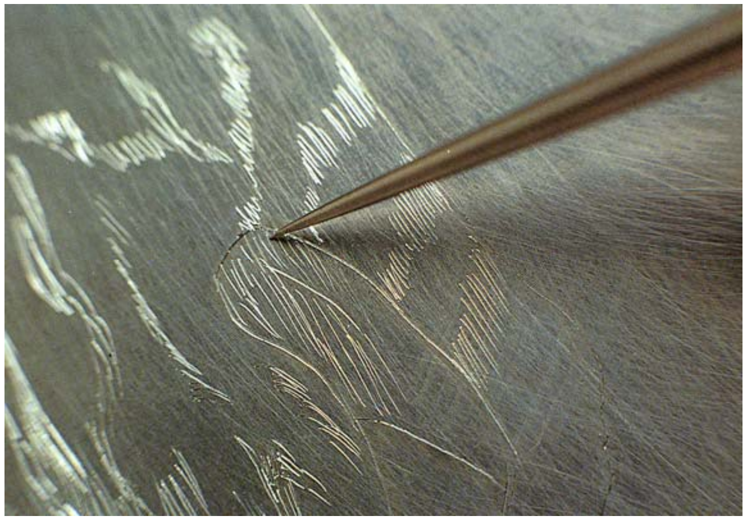


Figure 2: Engraving with a burin into the plate

Once the carving is complete, the plate is then covered with ink with a dauber or cloth, taking care to fill the grooves. While the ink is still wet, the surface of the plate is wiped clean, leaving only the ink in the recessed lines, as seen in Figure 3.

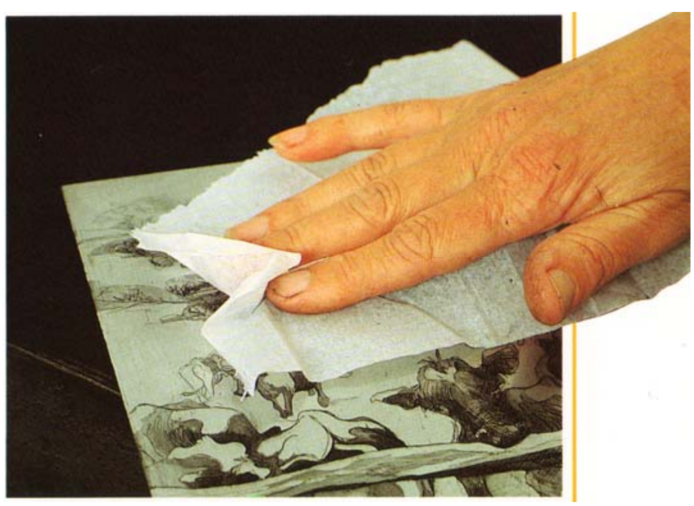


Figure 3: Wiping away the excess ink

To prepare the paper for the print, it is soaked, drained, and blotted. The plate is placed on the bed of the printing press, with the damp paper carefully placed on top of the inked surface. Felt blankets are placed over these two layers, providing a cushion so pressure is evenly applied. The press handle is then turned and the rollers of the printing press squeeze the plate and paper together, lifting the ink from the grooves onto the paper. The resulting print can be peeled off the plate and left to dry, as illustrated in Figure 4. [2]

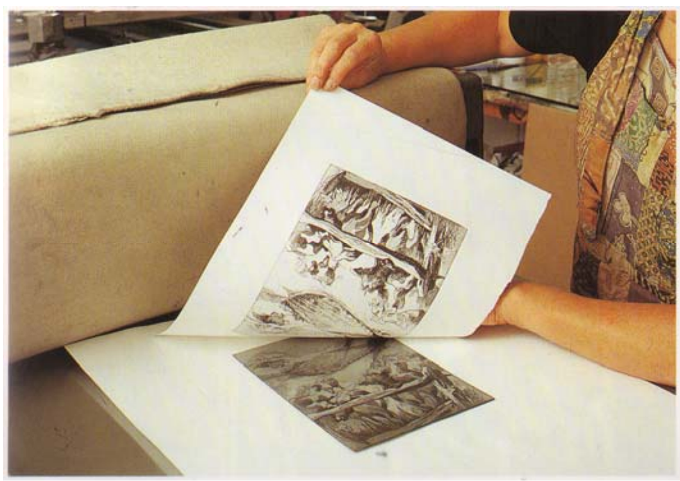


Figure 4: Peeling back the line-engraved print

In addition to black and white prints, intaglio prints can also include color. One method uses separate daubers for areas of the plate that should be different colors. Another technique makes use of multiple plates. Each plate has a section of the image that is inked with a different color. The plates are then printed on top of one another, similar to the printing method used by newspapers. [3]

Cartography of the Sixteenth Century

"Cartography is both an art and a science". [4] Sixteenth century mapmaking showcases the elaborate integration between geography and imagination. The mathematical precision and the inspired drawings resulted in a type of "decorative cartography" that mapped both familiar terrains and the unknown world.

Copperplate line-engraved maps are thought to have originated in Florence during the end of the 15^{th} century. These artistic maps had a distinct "map aesthetic" in regards to style, color, and text. The earliest maps printed bold, black lines against a contrasting stark uncolored white background. Another characteristic is the annotations in various places, designating the names of countries, waters, and other information, written in an efficient, legible print. [5]

Maps of this century used iconography that served "three effective roles: to decorate; to provide a useful symbol (for a church or house, for example); and, to form a foundation for other information." [6] This was the time of the Renaissance, and the illustrations intertwined the sciences, the arts, religion, and politics.

During the Renaissance, exploration in the sciences led to advancements in the fields of cartography, geometry, and astronomy. This knowledge provided a foundation for coordinate grid systems, used for positioning and geographic information. In 1569, Geradus Mercator, a Flemish cartographer and mathematician, produced the Mercator map projection for the purpose of nautical navigation. This type of projection "transforms the spherical surface [of the earth] to a plane so that a straight line on the resulting chart, anywhere in any direction, was a line of constant bearing." [7] Though this type of projection distorts size, it still retains shape and direction, and was used by early navigators to chart the their courses.

Universal symbols were used to distinguish certain geographical elements, such as hills, mountains, roads, and rivers. These symbols became more detailed with the introduction of copper plates. Waves now included different types of swirls; seas integrated a stippling pattern.

In addition to the necessary geographic information, these maps “never lacked ‘convenient spare and voide places’ to be filled with added ornament and little vignettes or thumbnail sketches” [1] (see Figure 5). Some elements of interest for this study include the whimsical fish, the fantastical sea creatures, the maritime vessels, and the movement and shape of the water waves.

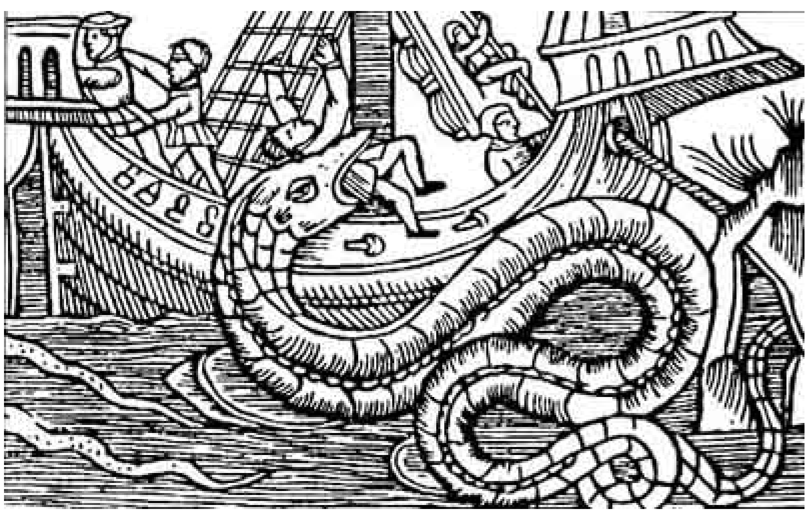


Figure 5: Sea dragon from “Historia de Gentibus Septentrionalibus” (1555) [8]

Some sketches were more than purely decorative. They also “conveyed information on national life: on custom, dress, and occupation in countryside or town; on natural resources and the methods of transporting them to market; on ships; [and] on instruments for both navigation and land survey.” [9] The religious and political iconography not only included images of kings and churches drawn over territories, but also more subtle symbolism of alliances and feuds. Many of these maps were produced for military reasons, and allusions were made to political alliances between regions. Several “hidden” meanings could be found in the drawings, especially those that had a critical or negative tone.

Copperplate maps began to incorporated color not only for aesthetic purposes, but also as a means of representation for symbols. According to Skelton, the following color associations were used in maps from the 15th century and later:

- Blues: Seas, Lakes, and Rivers
- Greens: Woods and Meadowlands
- Yellows, Browns: Roads
- Reds, Blues: Housetop Roofs
- Browns, Ochres: Mountains
- Specific tints for countries and provinces

Map Analysis

A historical and critical analysis of the Lafreri reproduction of the *Carta Marina* will help to establish some of the key criteria for this study. The specific areas of interest under evaluation are maritime elements.

Carta Marina Reproduction (1572). Olaus Magnus produced the original drawing of the *Carta Marina* in 1539. Nine woodblocks were used to print this multi-paneled Scandinavian map that illustrates the geography and folklore of Europe's Nordic region. With the panels assembled together, the *Carta Marina* was approximately four feet by five feet. Within the borders, this map overflows with sketches and illustrations of legends, politics, traditions, battles, and everyday life (see Figure 6). In 1572, Antonio Lafreri, a French mapmaker, created a reduced version of the *Carta Marina* using four copper plates. This line-engraved version contains much of the imagery found in the original. [10]

Figure 6: Close-up of Olaus Magnus' *Carta Marina* (1539)

A number of elements from the *Carta Marina* will be incorporated into this study, including the sea creatures, ships, and waves.

Several creatures can be seen in the water, such as lobsters, walruses, Dragons, monsters, and different species of fish. They are in various poses, with some partially submerged in the water and others intertwined with the ships. The features found on these multi-hued animals include scales, whiskers, fins, horns, and sharpened teeth.

Though small in size, the nautical ship drawings are quite detailed. The line work and color implies the ships are wooden. The shape of the ship is composed of several graceful, curved pieces, with intricate detail carved on the hull of the boat. Lines hanging from the mast support the many billowing sails. The ships in the Lafreri map slightly differ from the original. The copper-plate version has ships that resemble the Mediterranean style of ships at the time, versus Olaus' Swedish ships. These vessels appear "less happy under sail than Olaus' and…tended to be top-heavy." [11]

The currents can be seen in various swirls across the panels, adding to the movement of the map. The line work on the original *Carta Marina* fills the ocean with slightly wavy lines, but near Iceland, the lines transform into large, swirling strokes of water. These whorls may look to be a whimsical, artistic touch, but astonishingly, scientists have recently found that they "closely match a giant ocean front shown in satellite images" and correspond to large eddies in the water. It is believed that Olaus learned of this meteorological finding from sea mariners and included them in the map for navigational purposes [12]. Lafreri "infused billowy movement into his [reproduction] by innumerable little groups of curved lines, with shadows in the troughs of the waves." [11] The number, direction, and spacing of the lines enhance the stylized look of the water (see Figure 7). Also, these lines vary in thickness, giving an illusion of texture. This particular element of the study will pose a challenge in translating to 3D.

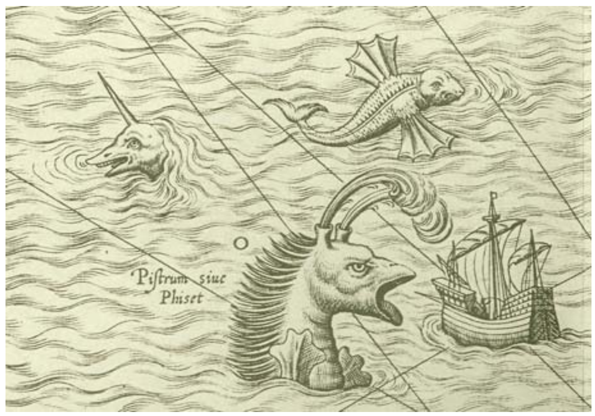


Figure 7: Close-up of Antonio Lafreri's *Carta Marina* (1572)

Other illustrated maps that will be examined include engravings found in the *Itinerario* and *The Mariner's Mirrour*. Many different engravers contributed to the *Itinerario*, which charts the journey to the Portuguese Indies [1]. Jan van Doetecom engraved *The Mariner's Mirrour*, and this sea atlas is comprised of over forty maps of the western European coastline [13]. Though these maps are not as elaborate or intricate as the *Carta Marina*, they do include many of the same characteristics significant to this study.

CHAPTER III

PRIOR WORK

The following publications and stylistic animated films provided a foundation to create a shader to be used in a non-photorealistic time-based animation.

Publications

Computer-Generated Pen-and-Ink Illustration. Winkenbach and Salesin discuss "how the principles of traditional pen-and-ink illustration…can be implemented as part of an automated rendering system." Principles of traditional hand drawings, such as stroke, tone, and texture, are implemented into a 3D pipeline. A stroke texture is a set of strokes that are used to create tone and texture seen in illustration. This concept was used as a foundation for re-creating the look of pen-and-ink for this system. [14]

Digital Facial Engraving. Ostomoukhov focuses on recreating digital facial engravings, which are based on the style of copperplate line-engraving. The sets of lines are created on separate layers which are then superimposed using various types of merging modes to create the final result. Additional enhancements, such as perturbing the engraved layers, further visually enhance the work. A simple color engraving process is also briefly discussed. [15]

Time Based Animations

Tale of How. This short animated film is part of a trilogy called "The Household", and was created by The Blackheart Gang in 2006 (see Figure 8). This hauntingly beautiful visual piece made use of traditional 2D animation integrated seamlessly with 3D animation. The entire work has a hand-drawn, artistic quality for the various elements. Part of the epic story takes place in the Indian Ocean, and a multi-layered approach was taken for the water waves. The layers of water were animated at various speeds and composited together, resulting in a 2.5D style. [16]

Figure 8: Image from Black Heart Gang's "Tale of How" (2006)

Le Régulateur. Philippe Grammaticopoulos' piece played at SIGGRAPH's Animation Theater in 2005 (see Figure 9). This unsettling tale follows a couple in search of creating the "perfect" child, though not without some obstacles. Non-photorealistic rendering is the focus of this high contrast, black and white animation. The line work gives the illusion of shape and form of the characters and environment. Lighting plays an important role as to how the lines are drawn across the surfaces. [17]

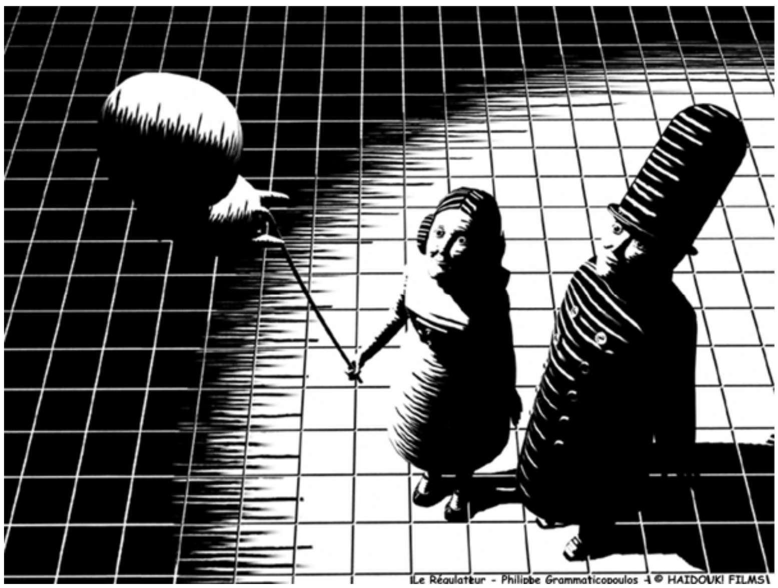


Figure 9: Image from Philippe Grammaticopoulos' "Le Régulateur" (2005)

CHAPTER IV

METHODOLOGY

General Technique for Copperplate Raytracing

Strothotte defines a general method for copperplate line-engraving [18] with the following steps:

1. t = intersection point as a result of a “ray query”
2. determine material properties at t
3. determine color (intensity) value at t for given illumination model
4. compute geometric and intensity parameters for hatching lines
5. post-processing to achieve the optical properties of copperplate images

The intersection point t is determined by a set of parallel planes that are overlaid objects in 3D space. Assume that point t = (x,y,z). Figure 10 shows the equation for the distance from a point to a plane is defined as:

$$distance = \frac{|Ax_1 + By_1 + Cz_1 + D|}{\sqrt{A^2 + B^2 + C^2}}$$

where at least one variable is not equal to zero

Figure 10: Equation for the distance from a point to plane

The variable co-efficients for this equation determine the orientation of the line sets, as illustrated in Figure 11. For example, an equation with co-efficients A = 0, B = 1, and C = 0 will create a set of vertical parallel lines. Each set of planes will result in a separate set of hatch lines.

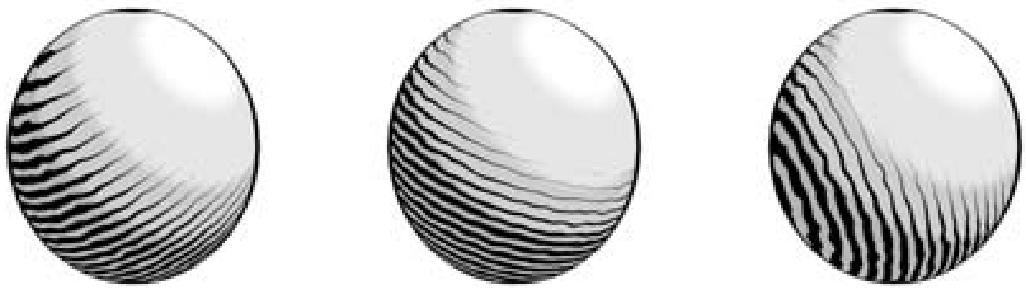


Figure 11: Spheres illustrating various line orientations

The result is a set of points representing a domain of values to be used for the striping function. This domain can then be scaled up or down to adjust the density of the lines, affecting the “level of detail”. Traditionally, objects that are closer to the eye will have more lines than those in the distance.

According to the technique described by Strothotte, these values are then fed into a piece-wise modulo equation, accounting for both the negative and positive values. However, this modulo function can instead be implemented using the floor() function, resulting in a faster, more efficient calculation. And because the distance will always be positive, only the first case of the piece-wise function needs to be addressed.

pattern = pattern – floor(pattern)

The output from this equation produces a saw-tooth function that varies between (0,1). The vertical regions of the saw-tooth function are clamped to create set of parallel lines on the surface of an object.

This process results in a basic pattern of lines that have several parameters that can be adjusted to further enhance the non-photorealistic look of copperplate line-engraving.

General Technique for Water Waves

The water illustrated in decorative maps is often done in a whimsical way, alluding to the movement of the currents and swirling eddies. These waves need to be translated to a 3D environment in a way that captures the look of the 2D maps. Imagery found in Japanese woodblock prints, such as “Sunset over Ryogoku Bridge” (1831-1833) by Katsushika Hokusai [19] can be used as visual inspiration for this piece (see Figure 12).

Figure 12: Hokusai’s “Sunset over Ryogoku Bridge” (1831-1833)

Typically, 3D water is done with a volumetric shader applied to the scene. This expensive calculation gives an illusion of depth, handling reflections, refractions, caustics, and visibility. Examples of such water can be found in Pixar's "Finding Nemo" (2003) and Dreamworks' "Shark Tale" (2004) (see Figure 13).

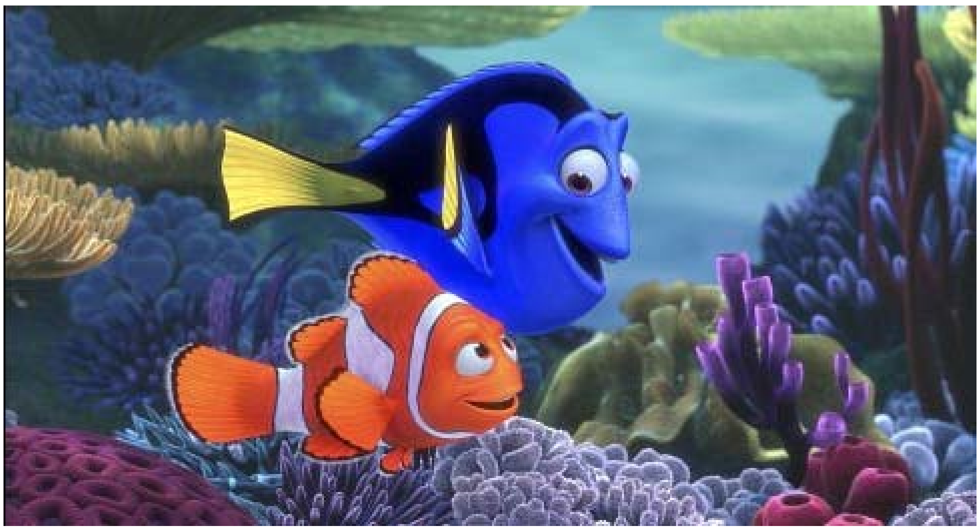


Figure 13: Image from Pixar's "Finding Nemo" (2003) [20]

However, this adaptation requires a very different look from the traditional 3D approach because it needs to give the appearance of being "flat", but still be readable from different angles. Several layers of water can be overlapped to give a look similar to the style seen in Blackheart Gang's 2006 animated piece "The Tale of How" [16], as seen in Figure 14.

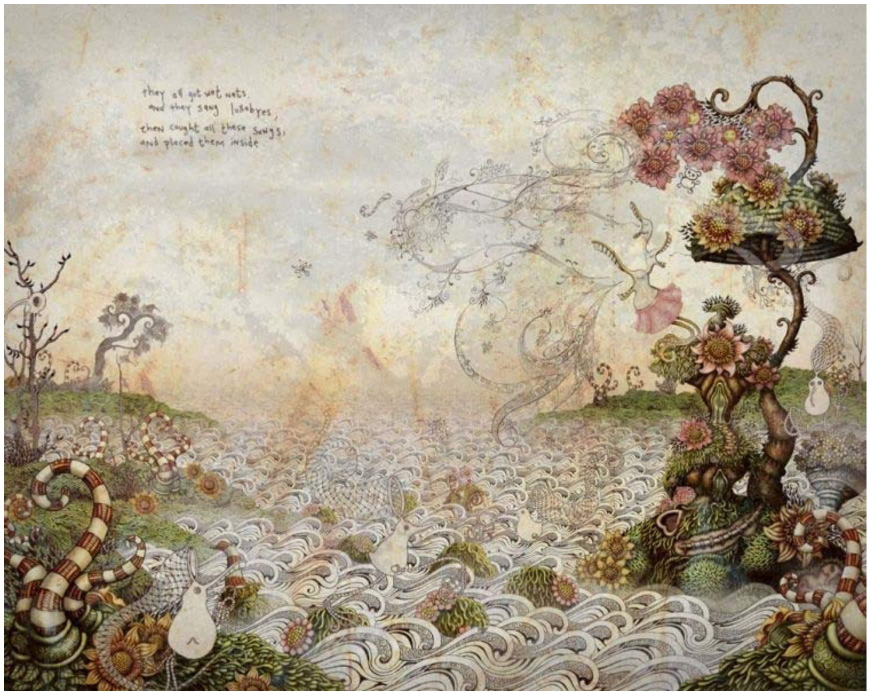


Figure 14: Artwork from Blackheart Gang's "Tale of How" (2006)

The visual style of the water waves can be managed either during the creation of the piece, or afterwards, in the post-production stage of the work, though both options have limitations. The waves could be modeled and animated in the scene file, and the characters can be integrated with the water. However, this approach leads to some limitations with the movement of the camera. A post-processing approach would be to composite water layers based on z-depth values. The renders would include a depth channel that would properly layer the water based on their assigned depth value. This would allow for greater flexibility with the animation of water, but may not integrate as well with the 3D geometry.

CHAPTER V

IMPLEMENTATION

To create a short animation with a similar look and style of 16^{th} century copperplate line-engraved maps, this implementation follows a simplified 3D production pipeline, focusing heavily on the shading and texturing aspects.

Models

The characters used for this time-based piece were polygonal models created in *Maya*™. Each model was UV unwrapped to allow for the option of adding additional texture-mapped details. Both models are from projects developed at Texas A&M's Visualization Laboratory. Audrey Wells assembled the intricate ship for her thesis "Virtual Reconstruction of a Seventeenth-Century Portuguese Nau" [21], and Kevin Singleton modeled and rigged the dragon for an independent project. The creature has a very organic shape, with many curves and bends on his body; the ship is very architectural and precise with its numerous details. The differences between these two models show adaptability of the copperplate shader.

Animation

The purpose of creating a time-based piece as opposed to still images is to show how the shader reacts over time. Poorly written shaders can have various problems, such as swimming across the surface of an object as it moves through space. Often times, even if the shader is correct, there can still be problems with how it looks because of other factors. For example, poor render settings with high shading rates and low pixel samples can lead to aliasing issues. For this particular shader, the animation movements cannot be extremely sudden, because that may cause the hatch lines to appear aliased or poorly drawn.

As the character animates, the movement of the skeleton causes the surface to deform. This poses a problem because the textures and shaders are dependant on the position of the model in 3D object space. An ideal solution was to create reference geometry for each surface mesh, providing a frame of reference for when the model deforms. Instead of shading point P, the built-in Pref variable was implemented as the source for the texture coordinates. This new variable corresponds to the shading point P on the reference model to calculate the correct shader values. [22]

varying point Pref = point “object” 0;

Lighting

Because the characters are moving through 3D space, the position of the lights is continually changing, altering the look of the line work and overall color. Lighting plays a critical role in the shader calculations that are based on the illumination values. The standard illumination model is defined as:

illumination = ambient light + diffuse light + specular light

The overall look for this piece does not have strong highlights so the specular component was disregarded. Because the ambient component evenly lights the entire surface, it also does not make a significant contribution to the desired aesthetic. For this reason, the illumination model was condensed to the diffuse component only. Assume Nf is the normal facing the camera, and Kd is the diffuse attenuation co-efficient. Then the illumination model can be defined as:

illumination = diffuse light = Kd * diffuse(Nf)

This equation is used for the calculations involving the shader parameters that are light-dependant, such as the tapering of the line and the regions of color variation. Because the lighting affects all of the line work, the lights were linked to the proper geometry to ensure the lines were drawn as expected.

Shading

Creating a pattern of computer-generated lines has a predictable, perfect look, whereas hand drawn lines have a number of imperfections. Because the goal of this copperplate shader is to calculate lines that look as if they have been drawn by hand, it is important to identify and implement characteristics typical of a hand-produced copperplate line-engraving.

When writing the shader, it is important to set up the correct coordinate system for the calculations to be applied correctly. *Renderman*™ uses various coordinate systems to describe predetermined geometric spaces. Points and vectors initially start out in “current” space, but it is critical to transform into “shader space” to ensure that the shader calculations stay in place, and do not slide or swim on the geometry.

A number of parameters must be adjusted to emulate the look of copperplate line-engraving. The parameters for this copperplate shader include the line width, wiggliness, tapering, outline, and fill color.

Line Width. The width of the line is based on clamping regions of the saw-tooth function, which can be increased or decreased by the user. If the result from the saw-tooth function is greater than the line width value, it is not considered to be a hatch line, and that area of the pattern is set to the fill color. But if it falls between 0 and the line width value, it is considered to be a line, and the region is set to the line color (see Figure 15). The greater the value for the line width, the thicker the hatch lines will be.

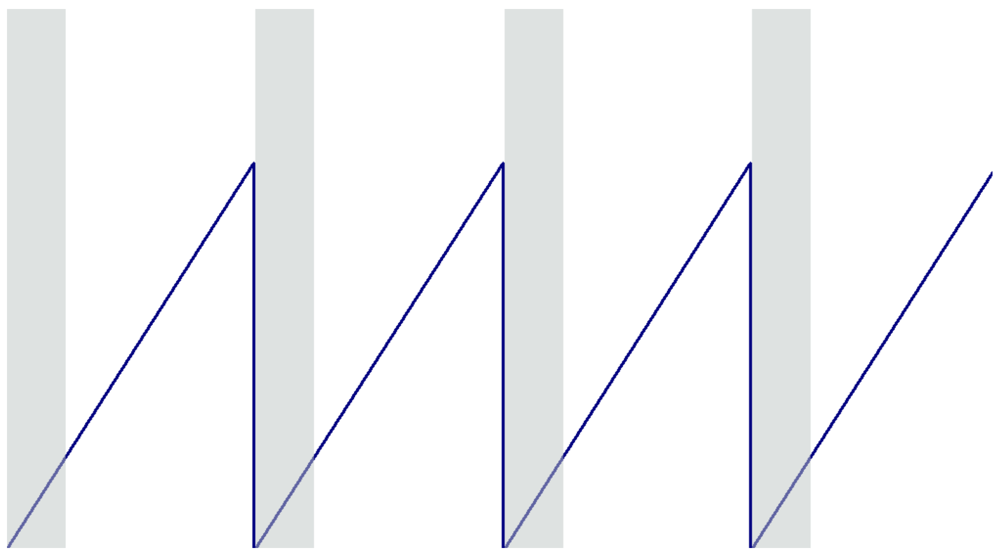


Figure 15: Clamping regions of the saw-tooth function

Wiggliness. The modulo function results in a set of perfectly straight, parallel lines. However, the lines in a copperplate illustration have a slight waver to them, because it is impossible for the human hand to remain perfectly steady as it is carving the design. In order to achieve this imperfection, noise is added to the shader point, so that it is slightly displaced from its original location. Figure 16 shows how increasing the "wiggliness" parameter affects the line.

```
wavy = vector noise(PP * 10) * wiggliness
PP = PP + wavy
```

Figure 16: Increasing "wiggliness" parameter value

Tapering. Tapering the line as it is drawn across the surface is achieved by varying the line width (see Figure 17). The region of the line that tapers falls within the user-defined start and end tapering points. Using a smoothstep() function, the width of line is mapped onto corresponding illumination levels. Because the line width is inversely proportional to the illumination values, it was reverse mapped to the illumination values to appear to taper.

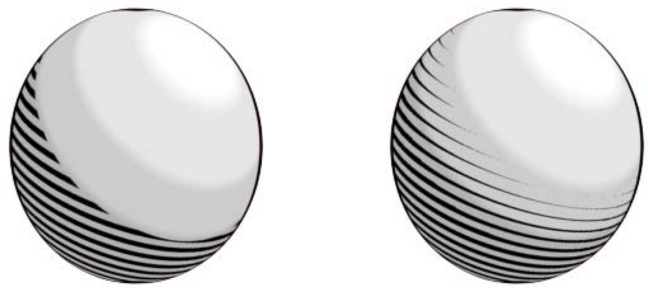


Figure 17: Tapering of the line

Outline. Sharp, bold outlines encompass the characters and objects found in copperplate maps. The initial approach to creating an outline around the geometry was to simply take the dot product of the surface normal and incident vector. If the two vectors are perpendicular the dot product will equal 0, detecting a silhouette edge. However, this led to poor, uneven results, especially noticeable around areas with high curvature.

In the second method, from Pixar's cell.sl shader, the CelOutline function modifies the above approach to create a cleaner silhouette (see Figure 18). Using derivatives creates a more even width outline. Adding the filterstep() function anti-aliases the edges of the outline. [22]

Figure 18: Apodaca's illustration of uneven and even width outlines

Fill Color. The fill color determines color regions within the silhouette. The approach is a slight variation from a traditional toon shader. Illumination values that fell within certain pre-defined ranges were clamped and assigned to one of the four distinct regions of color. This resulted in blocks of color that were separated with a hard, defined edge. To soften the boundary lines by creating a soft blend, the fill color between two regions was linearly interpolated using the illumination values. The different types of fill colors are illustrated in Figure 19.

Figure 19: Diffuse illumination, toon shading, and blended fill color

Texturing

Textures were used for two main components of this animation: water waves and the background image.

Different methods were attempted to translate the 2D water to a 3D world, and the best results were modeling and animating the waves in *Maya*™. Figure 20 shows how several rectangular-shaped planes of various heights were layered against one another, leaving small gaps between the planes to give an illusion of depth. These waves moved at staggered rates to create motion and make the water more dynamic. To transform the rectangles into waves, they were texture-mapped with an alpha mask, giving them a soft, curved shape, resembling drifting water, as seen in Figure 21. Though this approach smoothly integrated the water with the characters, it did limit the viewing angle of the camera. If the perspective was extreme, from the sides or looking down, it was obvious the water was not truly 3D, but only flat planes.

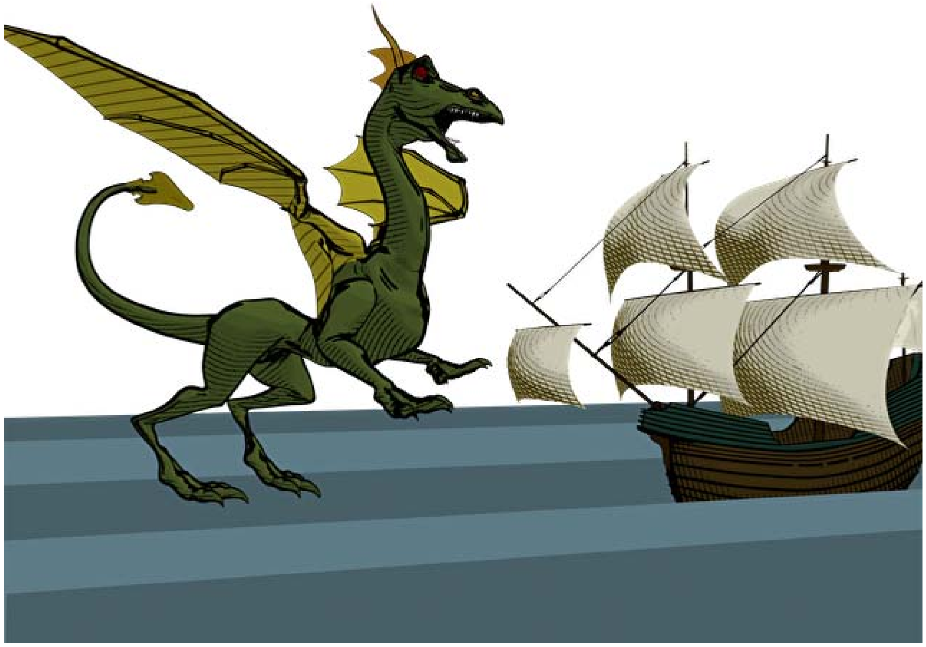


Figure 20: Rectangular “water” planes

Figure 21: Texture-mapped water waves

The background image needed to add to the overall feeling of the animation, but not be overly distracting. Based on research, because sea creatures were spotted far from civilization, the background should not include towns or houses. Instead, a simple texture of a faint compass was layered into the background. An additional paper texture was also incorporated, to give the background an aged textured look. This effectively enhanced the scene, without being overwhelming and taking away from the overall visual style.

Rendering

All images were rendered with very specific render settings to alleviate any aliasing problems, focusing on shading rate, pixel samples, pixel filter, and filter width (see Figure 22). The shading rate determines how many times a shading computation is calculated per given region of an image. For this project, setting this value at 0.20 means that every pixel is shaded 0.20 pixels in each direction, or a total of 25 times per pixel. Pixel samples reduce aliasing problems by supersampling. Because of the fine detail in the line work, this option was set to production quality at 11x11 samples. Though using the sinc pixel filter is computationally expensive, this filter works best for hard lines and edges. Further information on the additional rendering settings can be found in the *Renderman*™ documentation.

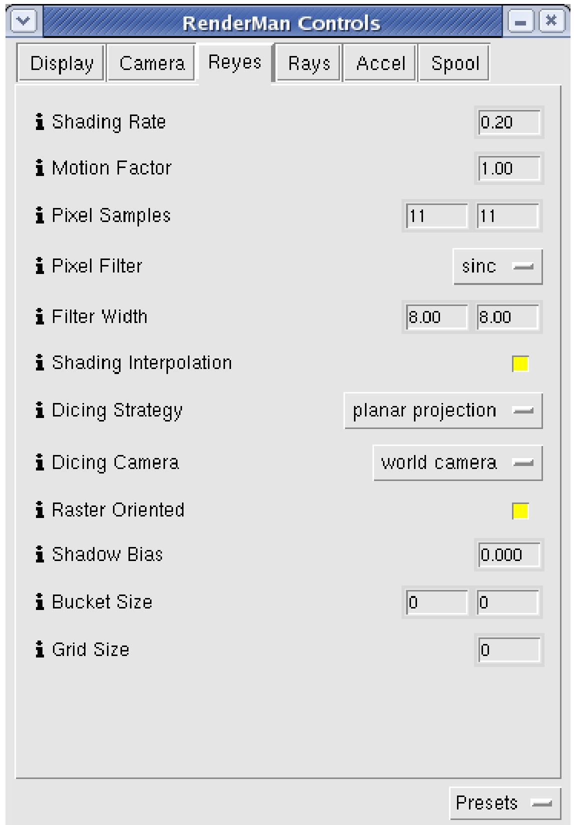


Figure 22: Render settings

Post Process

Post-processing techniques further enhanced the look of the time-based animation to resemble the look of a 16th century map. These techniques were applied to the entire animation to give it a more cohesive feel.

As aged paper texture was composited in as the background layer, and it was processed to fit the overall look of the piece. Because copperplate line-engraved maps from this time were pressed onto handmade paper, it had a slightly textured look. To re-create this visual characteristic, a blank surface texture was created in *Adobe*

Photoshop™ and overlaid onto the animation sequence using Apple's *Final Cut Pro*™. The texture gave a slight impression of a paper-like quality, but was not overly distracting to the viewer.

In addition to overlaying a paper texture, color adjustments were also applied. The colors were carefully selected in *Renderman*™ during the shading process, but subtle changes helped to better match the color quality from maps of the 16th century. The color values were tweaked to give a warmer, slightly yellow tinge to the sequence, as well as lowering the overall saturation (see Figures 23 and 24).

Figure 23: Before color correction process

Figure 24: After color correction process

CHAPTER VI

RESULTS

The final product from this research is a short, time-based piece implementing a non-photorealistic, copperplate line-engraved shader. The objective was not to create a direct translation of these maps, but rather an adaptation of this 2D style to a 3D environment. The line quality, color, and subject matter are all suggestive of illustrative maps from the 16^{th} century.

The results demonstrate use of different line quality and use of color. The line strokes are the most distinctive aspect of traditional copperplate engravings. Numerous parameters give the user control over subtle variations in line quality. Parameters like wiggliness, line width, and tapering are all characteristic of traditional line engravings. These subtleties give the overall image visual enhancement and depth.

Additional elements, like silhouettes and color, further enhance the richness of the piece. Because strong, heavy contour lines are typical of traditional decorative maps, bold, black outlines were applied to the 3D models. As an artistic decision, the hatch lines were a darker version of the fill colors, as opposed to the black line work often seen in cartography from that time. The color selections were muted and low in saturation, to keep to the aesthetic of the 16^{th} century maps.

The implementation and results were successful in terms of adapting the look of copperplate line-engraving to 3D. Figures 25-27 are renders from the time-based animation present the results that have been achieved with the copperplate line-engraving shader. Additional images illustrating the application of the copperplate shader can be found in Appendix A.

Figure 25: Frame from final animation

Figure 26: Close-up render of dragon

Figure 27: Close-up render of ship

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

Future Work

Though this thesis achieved its goals, it does have the potential to be expanded in a number of ways, either within the actual shader or on a larger scale.

The shader code can be further developed to enhance the look of copperplate line-engraving. The current code includes only certain parameters but a number of others could be implemented. One additional parameter that may be useful would be randomizing the length of lines by a small distance, since a set of hand drawn lines do not all end at the same point.

Another possible modification to the shader would be adjusting how it responds to light. Ideally, not all lights in the shader would alter the line quality. Instead, designated lights would create the line work, and additional lights would be used for backlighting or other traditional purposes.

Currently, the fill color for the objects is clamped to create four softly blended regions. The change in value between these regions is explicitly set, and the changes in value are subtle. Additional modifications to the code could enable the user to decide the amount of change between these regions, giving a more dramatic variation between boundaries.

The code could also be developed to better handle the problems of spatial aliasing, exploring additional filtering and sampling options.

This short animation illustrated how the shader held up over time and across space. But, the copper-engraving method adapted in this thesis can also be applied to a full length animation, book illustrations, and other media.

Finally, this work can be used in a collection of *Renderman*^{TM} shaders involving relief and intaglio printmaking. Each style has certain characteristics, and this shader could provide a basic framework to be expanded upon. For example, alternate printmaking styles used for mapmaking can be developed, including woodcuts of the 15th century or lithographic maps dating from the 19th century.

Conclusions

In conclusion, this thesis successfully developed a method to adapt the visual style of 16th century copperplate line-engraved maps to a 3D environment. A number of parameters provided flexibility to produce images that were both accurate and visually pleasing. The resulting 3D animation captured the look and style of this unique 2D printmaking method.

REFERENCES

[1] R. A. Skelton, *Decorative Printed Maps of the Fifteenth to Eighteenth Centuries: A Revised Edition of Old Decorative Maps and Charts by A. C. Humphries*, London: Staples, l952.

[2] J. Ross and C. Romano, *The Complete Intaglio Print*, New York: The Free Press, MacMillan, 1974.

[3] J. Martin, *The Encyclopedia of Printmaking Techniques*, Philadelphia: Running Press, 1993.

[4] J. B. Krygier, "Cartography as an art and a science," in *Cartographic Journal*, vol 32, no 1, 3-10.

[5] D. Woodward, *Maps as Prints in the Italian Renaissance: Makers, Distributers, and Consumers*, London: Henry Ling (Printers) Ltd, 1996.

[6] M. Swift, *Mapping the World*, New Jersey: Book Sales, 2006.

[7] A.H. Robinson, J.L. Morrison, P.C. Muehrcke, A.J. Kimerling, and S.C. Guptill, *Elements of Cartography* Sixth Edition, New York: John Wiley & Sons, 1995.

[8] "Category:Historia de gentibus septentrionalibus" http://commons.wikimedia.org/wiki/Category:Historia_de_gentibus_septentrionalibus. Accessed February 10, 2008.

[9] C. Bricker, *Landmarks of Mapmaking: An Illustrated Survey of Map and Mapmakers. Maps chosen and displayed by R.V. Tooley; text written by Charles Bricker; preface by Gerald Roe Crone*, The Netherlands: Elsevier, 1968.

[10] "Bell Library: Maps and Mapmakers," http://bell.lib.umn.edu/map/OLAUS/indexo.html. James Ford Bell Library: University of Minnesota. Accessed June 08, 2007.

[11] E. Lynam, *The Carta Marina of Olaus Magnus, Venice 1539 & Rome 1572*, Pennsylvania: Tall Tree Library, 1949.

[12] BBC News: "Ancient map captures ocean front." http://news.bbc.co.uk/2/hi/science/nature/3683087.stm. BBC News Science/Nature. Accessed September 1, 2007.

[13] G.R. Crone, "The Mariner's Mirrour 1588", *Geographical Journal*, vol 119, 455-458, December 1953.

[14] G. Winkenbach and D. H. Salesin, "Computer-generated pen-and-ink illustration" in *Computer Graphics*, Annual Conference Series, vol. 28, pp. 91-100, 1994.

[15] V. Ostromoukhov, "Digital facial engraving," in *Computer Graphics Proceedings,* Annual Conference Series, pp 417-424, 1999.

[16] 'The Tale of How - The Black Heart Gang," http://www.thebleackheartgang.com/ Accessed February 10, 2008.

[17] "Le Régulateur – un film de Philippe Grammaticopoulos," http://www.zewebanim.com/regulateur.htm. Accessed July 30, 2007.

[18] S. Strothotte and S. Schlechtweg, *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*, San Francisco: Morgann Kaufman, 2002.

[19] J. M. White, R. M. Brandon, and Y. Woodson, *Hokusai and Hiroshige: Great Japanese Prints from the James A. Michener Collection, Honolulu Academy of the Arts,* Asian Art Museum of San Francisco, 1999.

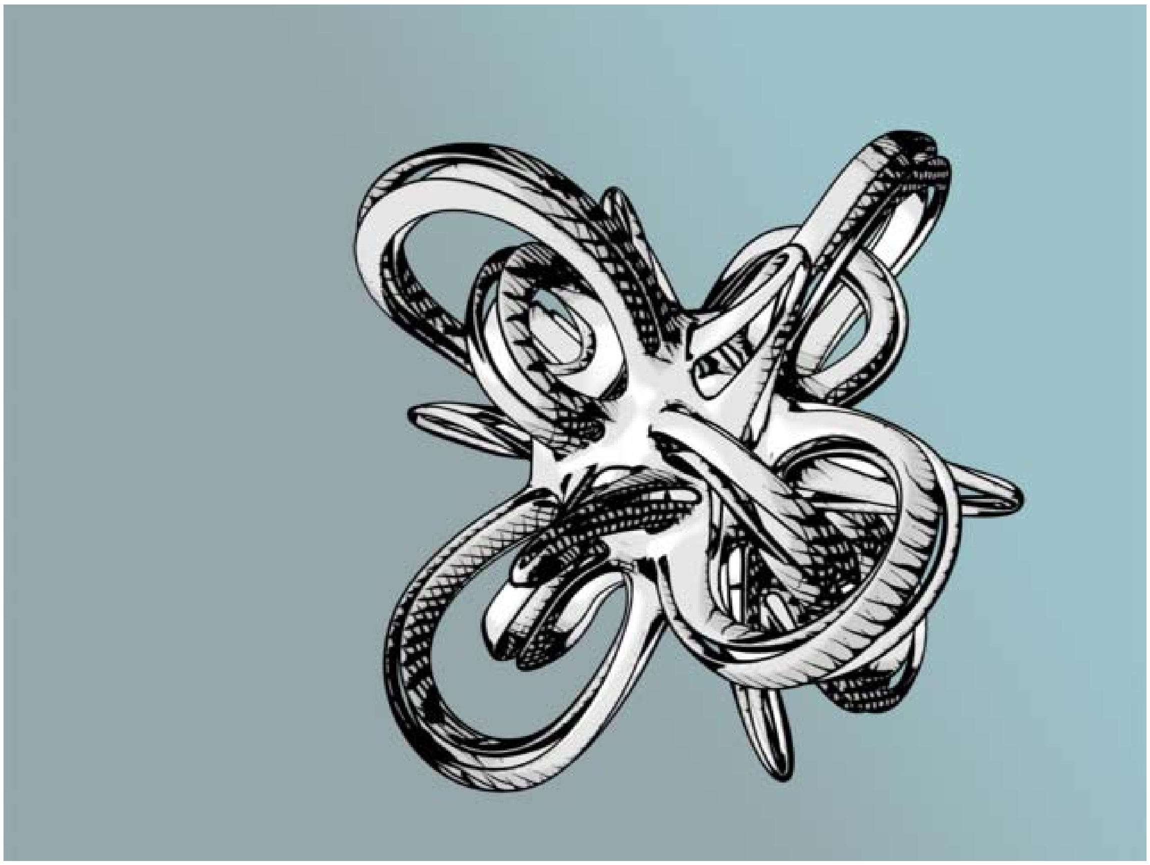
[20] "IMDB- Photos from Finding Nemo" http://www.imdb.com/media/rm1646893056/tt0266543. Accessed February 10, 2008.

[21] A. Wells, "Virtual Reconstruction of a Seventeenth-Century Portuguese Nau," M.S. thesis, Texas A&M University, 2008.

[22] A. A. Apodaca, L. Gritz , and B. A. Barsky, *Advanced RenderMan: Creating CGI for Motion Picture*, San Francisco: Morgan Kaufmann Publishers Inc., 1999.

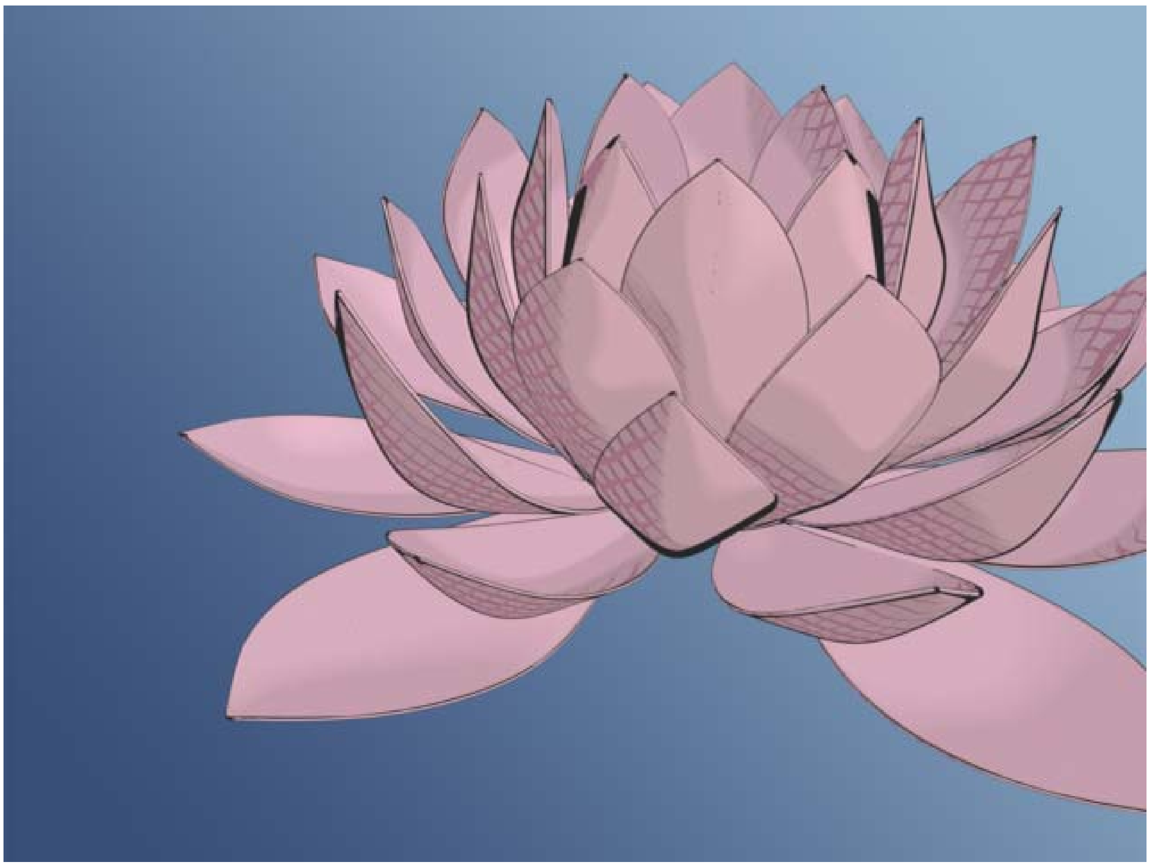
APPENDIX A

The following image gallery illustrates the application of the copperplate shader. The models chosen represent a variety of shapes and forms, demonstrating adaptability of the shader.



Copperplate shader applied to symmetrical model

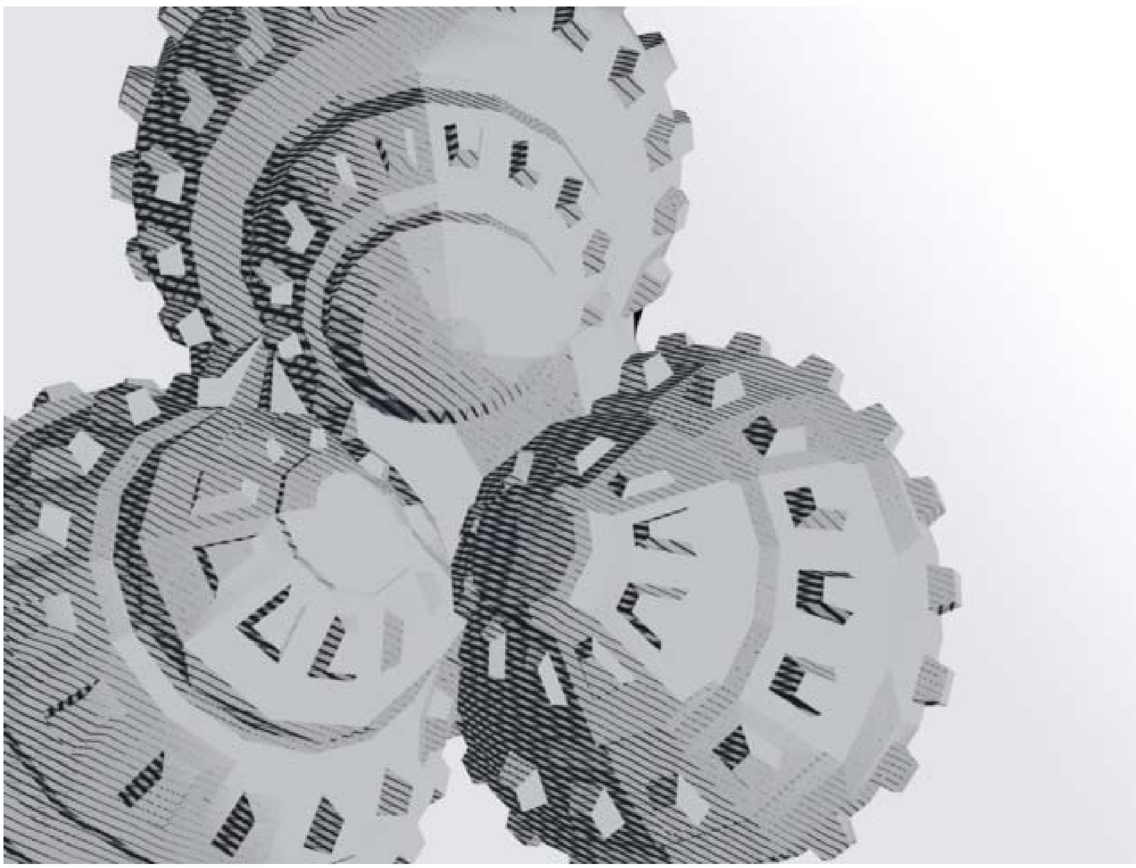
Model by Audrey Wells



Copperplate shader applied to lotus flower

Model by Katherine Van Maanen

Copperplate shader applied to bass



Copperplate shader applied to gears

Model by Katherine Van Maanen

APPENDIX B

```
//  Shaila Haque
//  Texas A&M University
//  MS Visualization Sciences 2008
//  Renderman™ shader code for copperplate.sl

#include "headers/patterns.h"
#include "headers/filterwidth.h"

/* outer edge shading calculations*/
/* Pixar toon shader */
color outlines( normal N; normal V;
                float outlineThickness; color outlineColor; color baseColor)
{
  float angle = abs(normalize(N) . normalize(I));
  float dangle = filterwidth(angle);
  //scale the angle to control the threshold near curvature
  float outerEdge = 1 - filterstep (10 * outlineThickness,angle/dangle);
  return mix(baseColor, outlineColor, outerEdge);
}

/*color variation*/
color variation(color backgroundColor; float illumination)
{
  color colorVariation = backgroundColor * illumination;
  return colorVariation;
}

/* creates hatch line sets */
color hatch(    varying point __Pref; color colorVariation; color hatchColor;
                float xVal; float yVal; float zVal;
                float lineWidth; float lineDensity; float wiggliness;
                float taperStart; float taperEnd;
                color diffuse; float illumination)
{
  point PP =  transform("object", __Pref);
  // add noise for a wavy line
  // point + vector = point
  vector wavy = 0;
  wavy = vector noise(PP * 10) * wiggliness;
  PP = PP + wavy;
```

```
  color hatchOutput;

  //extract x,y,z components of shading point P
  float x = xcomp(PP);
  float y = ycomp(PP);
  float z = zcomp(PP);

  //plane1 direction, where a, b, c represent the co-effients of the plane
  float a = xVal;
  float b = yVal;
  float c = zVal;
  float d = 0;

  //equation for distance from point to a plane
  float distance = ((a*x) + (b*y) + (c*z)) / sqrt ((a*a) + (b*b) + (c*c));

  //create striped pattern, adjusting the number of lines in the pattern
  float pattern = distance * lineDensity;
  pattern = pattern - floor(pattern);

  //set clamping values to create pattern of hatch lines
  if (pattern < lineWidth)
  {
    if (illumination < taperStart)
    {
      hatchOutput = hatchColor;
    }
    else if ((illumination > taperStart) && (illumination < taperEnd))
    {
      float taper = lineWidth * (1 - smoothstep(0.0, lineWidth, illumination));
      if (pattern < taper)
        hatchOutput = hatchColor;
      else
        hatchOutput = colorVariation;
    }
    else if (illumination > taperEnd)
    {
      hatchOutput = colorVariation;
    }
  }
  else
    hatchOutput = colorVariation;
  return hatchOutput;
}
```

```
surface
  copperplate(
                    float outlineThickness = 1.5;
                    color outlineColor = color(0,0,0);
                    color backgroundColor = color (1.0, 1.0, 1.0);

                    float x1 = 1;
                    float y1 = 3;
                    float z1 = 1;
                    color hatchColor1 = color (0,0,0);
                    float p1lineWidth = 0.55;
                    float p1lineDensity = 12;
                    float p1wiggliness = 0.04;
                    float p1taperStart = 0.2;
                    float p1taperEnd   = 0.55;

                    float x2 = 5;
                    float y2 = 2;
                    float z2 = -3;
                    color hatchColor2 = color (0,0,0);
                    float p2lineWidth = 0.45;
                    float p2lineDensity = 16;
                    float p2wiggliness = 0.03;
                    float p2taperStart = 0.15;
                    float p2taperEnd   = 0.6;

                    varying point __Pref = point "object" 0;
                    )
{
  /* Nf: normalized vector that faces toward the camera */
  /* I: vector from camera towards the surface point */
  /* V: normalized vector from surface point to camera */
  normal Nf = faceforward(normalize(N),I);
  vector V = -normalize(I);

  // lighting & illumation calculations
  float illumination = 0;
  color ambient, diffuse, specular;
  float Ka = 0.6, Kd = 0.8, Ks = 0.4;
  float highlight = 1.0;
  float roughness = 0.1;

  ambient = Ka * ambient();
```

```
diffuse = Kd * diffuse(Nf);
specular = Ks * specular(Nf, V, roughness) * highlight;

// illumination equation
illumination = comp(diffuse, 0);

float clampIllumination = 0;


// y = mx + b
// example: m = slope = (1.0 - 0.9)/(0.75 - 0.70) = 2

if (illumination > 1.0)
  clampIllumination = 1.0;

if(illumination < 1.0 && illumination >= 0.75)
  clampIllumination = 1.0;

if (illumination <0.75 && illumination >= 0.50)
{
  if (illumination < 0.75 && illumination >= 0.70)
  {
    //blend between 0.9 and 1.0
    clampIllumination = 2.0*illumination - 0.5;
  }
  else
    clampIllumination = 0.9;
  }

if (illumination <0.50 && illumination >= 0.25)
{
  if (illumination < 0.50 && illumination >= 0.45)
  {
    //blend between 0.8 and 0.9
    clampIllumination = 2.0*illumination - 0.1;
  }
  else
    clampIllumination = 0.1;
  }

if (illumination <0.25 && illumination >= 0.00)
{
  if (illumination < 0.25 && illumination >= 0.20)
  {
```

```
      //blend between 0.7 and 0.8
      clampIllumination = 2.0*illumination + 0.3;
    }
    else
      clampIllumination = 0.7;
    }

  Oi = Os;

  Ci = variation(backgroundColor, clampIllumination);
  Ci = hatch(__Pref, Ci, hatchColor1, x1, y1, z1,
             p1lineWidth, p1lineDensity, p1wiggliness,
             p1taperStart, p1taperEnd, diffuse, illumination);

  Ci = hatch(__Pref, Ci, hatchColor2, x2, y2, z2,
             p2lineWidth, p2lineDensity, p2wiggliness,
             p2taperStart, p2taperEnd, diffuse, illumination);

  Ci = outlines(N, V, outlineThickness, outlineColor, Ci);
}
```

VITA

Shaila Haque

Texas A&M University
C418 Langford Center
3137 TAMU
College Station, TX 77843-3137

shaila.haque@gmail.com

Education

B.S., Computer Science, Texas A&M University, 2004
M.S., Visualization Sciences, Texas A&M University, 2008